```
% Example of SVM testing and training using OSU_SVM toolbox
% Solves XOR problem
clear all;  % Clear all variables from memory

X = [-1 +1 -1 +1;  % XOR
     -1 -1 +1 +1]; % Data
Y = [-1 +1 +1 -1]; % Targets

N = length(Y);  % Number of samples

%% Normally data is split into 5 folds (this has not been done here!)
% Define training data
test_range = 1:N;
train_range = [1:N];
train_data = X(:,train_range);
train_class = Y(train_range);
%Define testing data
test_data = X(:,test_range) ;
test_class = Y(test_range);

%%  Define SVM Kernel Hyperparameters
C = 10;  % Define C (box constraint)
G = 2;   % Define Gamma = 1/Sigma (width of Rbf or order of polynomial)
%% "Grid search" ideally needed but not included here

%% Train the Linear kernel SVM (comment out the other)
%[AlphaY,SVs,Bias,Parameters,nSV,nLabel] = LinearSVC(train_data,train_class,C);

%% Train the Polynomial kernel SVM (comment out the other)
[AlphaY,SVs,Bias,Parameters,nSV,nLabel] = PolySVC(train_data,train_class,G,C);

%% Train the RBF kernel SVM (comment out the other)
%[AlphaY,SVs,Bias,Parameters,nSV,nLabel] = RbfSVC(train_data,train_class,G,C);

%% Test output (cross-validation not used here - must add!)
[ClassRate, DecisionValue, Ns, ConfMatrix, PreLabels]= SVMTest(test_data,
test_class, AlphaY, SVs, Bias, Parameters, nSV, nLabel);

%% Plot results
clf;figure(1);hold;
for n=1:N,
    if PreLabels(n)==1,

plot(X(1,n),X(2,n),'bx','markersize',10,'linewidth',2,'markerfacecolor','b');
    else

plot(X(1,n),X(2,n),'ro','markersize',10,'linewidth',2,'markerfacecolor','r');
    end
end
axis('square');grid
axis([-2 2 -2 2]);
line([-2 2],[0 0],'LineWidth',1.5,'Color',[0 0 0]);
line([0 0],[-2 2],'LineWidth',1.5,'Color',[0 0 0]);
xlabel('x_1','fontsize',16);
ylabel('x_2','fontsize',16);

%% Only points plotted here, best to include the optimal separating
%% hyperplane and canonical planes if you can!!!
```